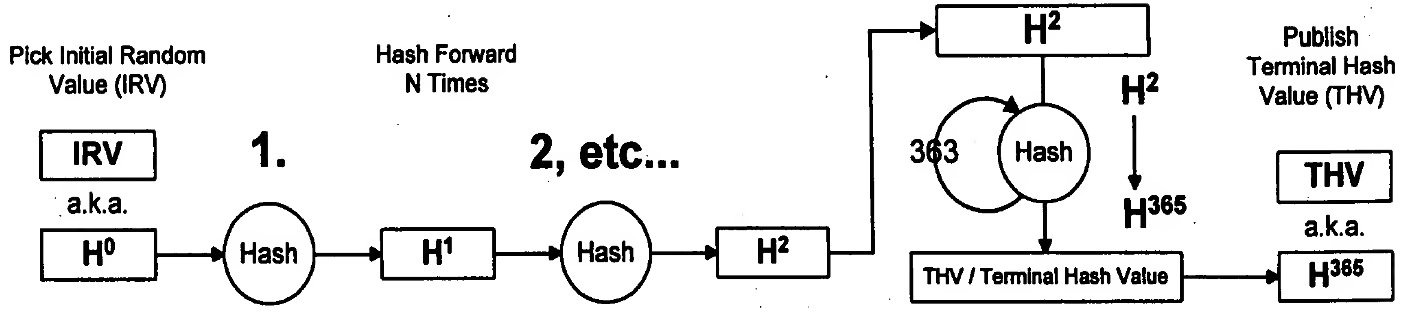
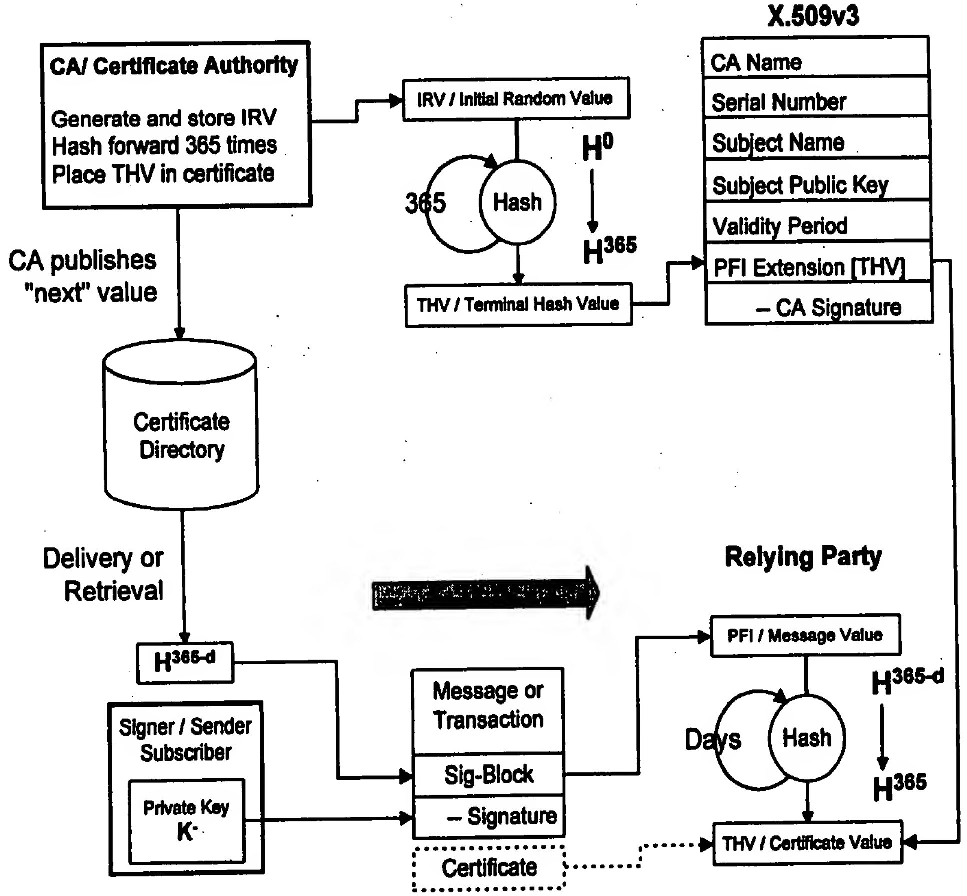
What is Iterated Hashing?


Pick Initial Random Value (IRV)
IRV
a.k.a.
H⁰
1.
Hash
Hash Forward N Times
H¹
2, etc...
Hash
H²
H²
363
Hash
H²
H³⁶⁵
THV / Terminal Hash Value
Publish Terminal Hash Value (THV)
THV
a.k.a.
H³⁶⁵

- A simple, fast, secure way to prove you know a secret.
- Hash functions are “fast” but infeasible to reverse.
- Pick a random number (IRV), hash it N times ($H^0 \ldots H^N$), and then publish the terminal value ($THV = H^N$).
- Revealing a “prior” value (H^{N-D}) proves you know IRV.
- Recipient verifies by hashing forward D times to THV.
- Effect (on a cert) is similar to issuing a fresh signature.

Fig. 1

Application to Secure E-Mail



Sender retrieves PFI value
Places in signature block
Signs and sends message

Receive and verify signature
Extract PFI from sig block
Hash forward D days
Compare with THV from cert

Can also support:

- 2-Hourly
- Hourly
- 30 Minutes
- 10 Minutes

PFI = periodic freshness indicator

1 public key operation is equivalent to (approx.) 10,000 hash operations.

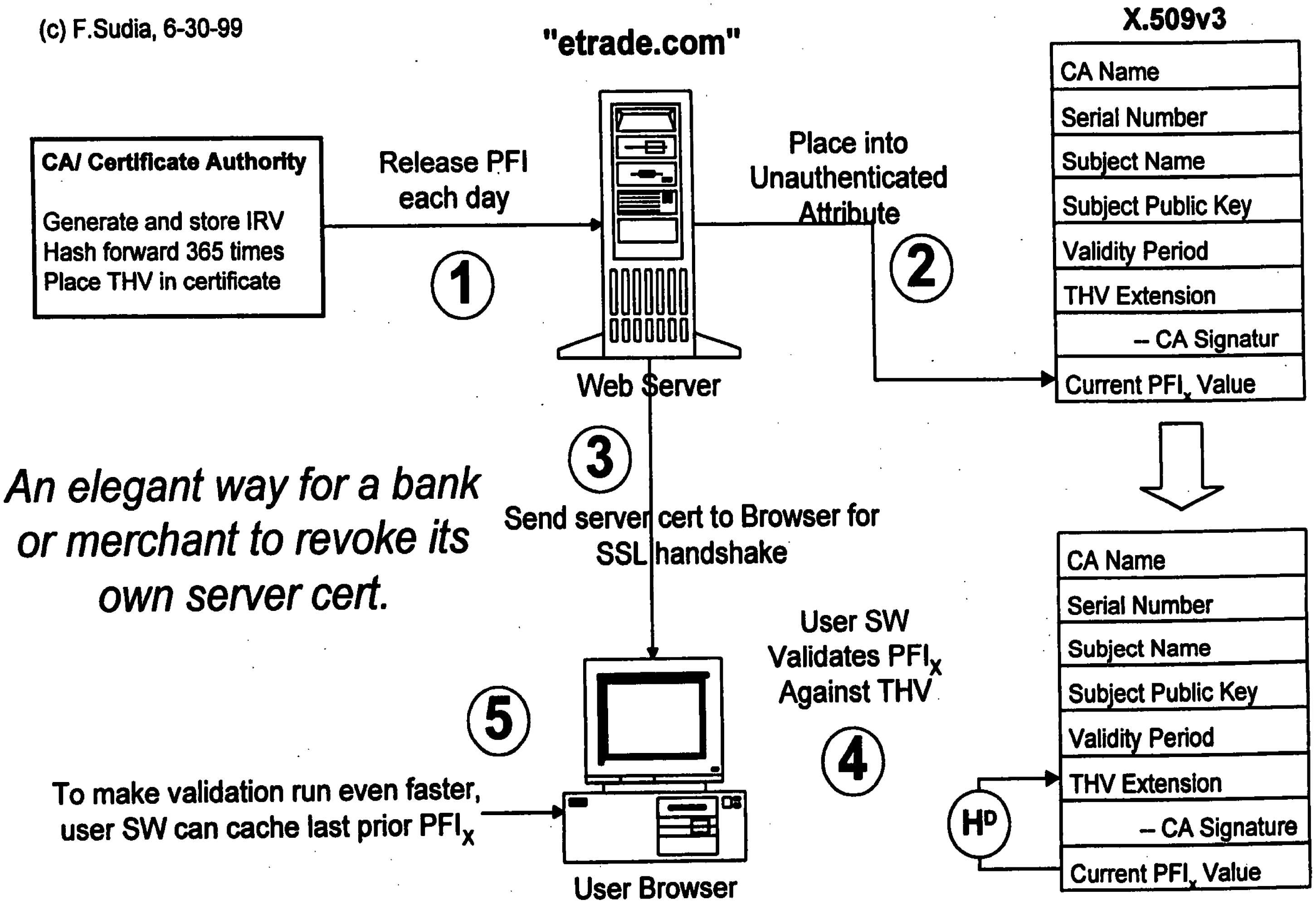
Recipient performs no online communication.

(c) F.Sudia, 6-28-99

Fig. 2

Application to Server Certs

(c) F.Sudia, 6-30-99



"etrade.com"

CA/ Certificate Authority

Generate and store IRV
Hash forward 365 times
Place THV in certificate

1 Release PFI each day

2 Place into Unauthenticated Attribute

3 Send server cert to Browser for SSL handshake

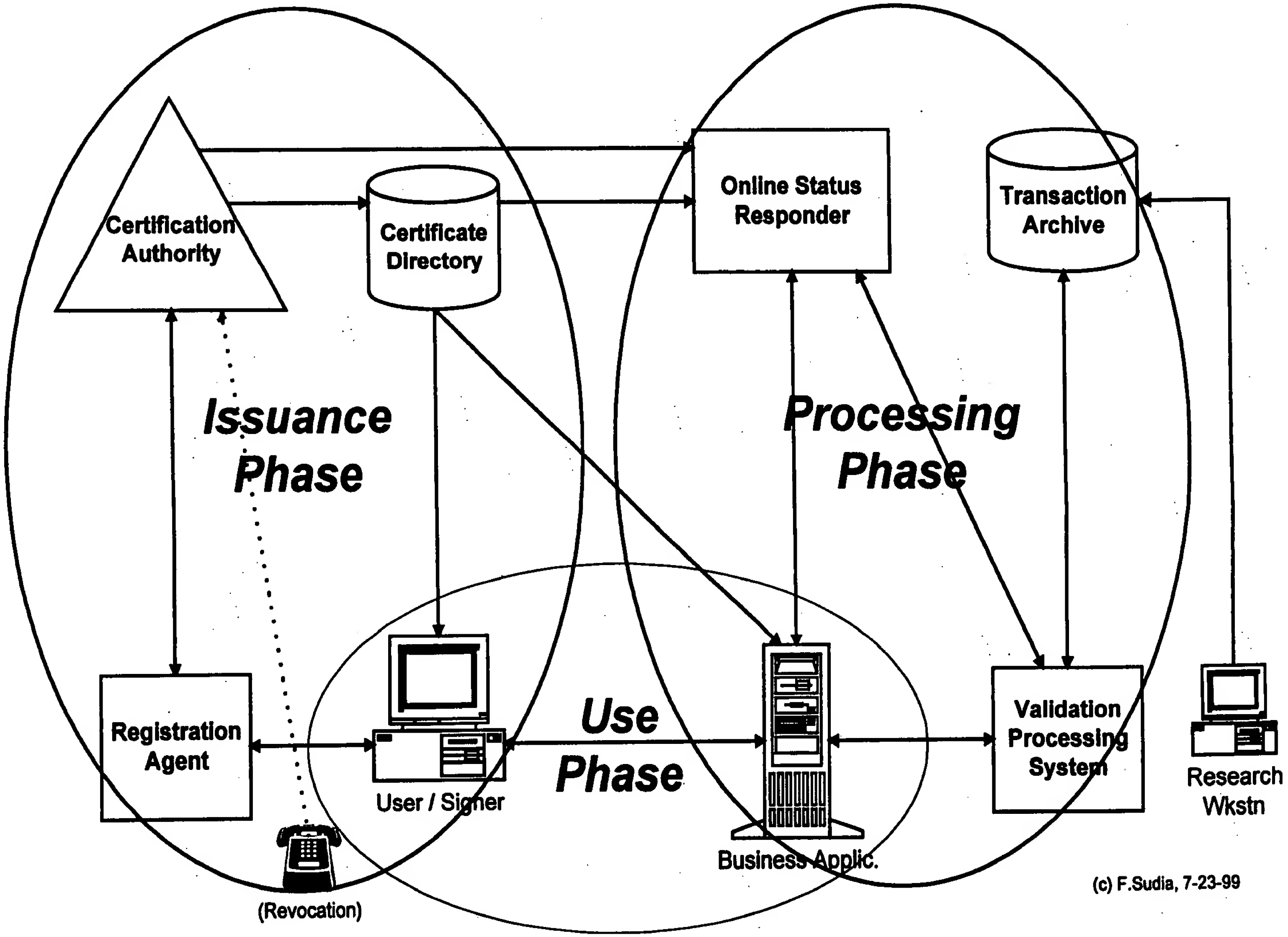
4 User SW Validates PFI_X Against THV

5 To make validation run even faster, user SW can cache last prior PFI_X

An elegant way for a bank or merchant to revoke its own server cert.

Fig. 3

Structure of Market



(c) F.Sudia, 7-23-99

Fig. 4